Association for Information Systems
AIS Electronic Library (AISeL)

MCIS 2017 Proceedings

Mediterranean Conference on Information Systems (MCIS)

9-2017

Remote Tool For Controlling User Interface Of Mobile Applications

Maya Stoeva
University of Plovdiv Paisii Hilendarski, may_vast@yahoo.com

Nikolay Pavlov
University of Plovdiv Paisii Hilendarski, npavlov@kodar.net

REMOTE TOOL FOR CONTROLLING USER INTERFACE OF MOBILE APPLICATIONS

Research full-length paper

Track N°8

Stoeva, Maya, University of Plovdiv Paisii Hilendarski, Faculty ot Mathematics and informatics, may_vast@yahoo.com

Pavlov, Nikolay, University of Plovdiv Paisii Hilendarski, Faculty ot Mathematics and informatics, npavlov@kodar.net

Abstract

It will be very useful for users to have an opportunity to control external visual appearance of their applications, especially the mobile one. This potentially functional feature provoked the authors beginning a research on development of mediator tools, which can help users for that. In this paper, will be presented a flexible tool, that controls remotely and dynamically visual appearance of cross platform mobile native shopping applications. Its architecture is multi-layered and later can be used like a base extension for other connecting tools. They can manage UI mobile applications from different types.

Keywords: Multi-layered architecture, User interface (UI), UI Remote tool, Native mobile application, iOS, Android, YML, PHP Symfony, MySQL, Response-Request Interactions, MVC model, Responsive design, User Experience (UX), Usability.

1 Introduction

The growth of mobile influence increases every day. People often use their mobile devices for activities and services such as online shopping, organizing meetings, showing presentations and managing their bills. The number of applications, developed for these purposes, also becomes bigger and requirements for their user interfaces (UI) – too. Mobile software follows different design corporate identity (CI), which is visualized on different kinds of operating systems.

The presented tool is flexible supporting software from backend type and is developed like a native terminal application for mobile devices, working on iOS and Android operating systems. The described solution can be visual fully customized.

This software inherits and expands the original idea of Interactive Multimedia tool for dynamic generation of web interfaces (Stoeva, 2014). In paper (Stoeva, 2014) is described why one of the main components for every software system is its user interface. The focus is on UI wireframing and prototyping – two great techniques for creating and checking a lot of ideas from different project stakeholders. This interactive tool (Stoeva, 2014) allows users to build and to present an interface vision of a final product, which can be used later from the programmers. Thereby they can save time and avoid wrong ideas, connected with the future application development. When this sketching tool was developed, and going into use, the authors realized that is not only enough to clarify design in a begging project stage. Because often customers need to change something in the UI, after the software is ready. In summery that situation, could be raise up from different reasons and stakeholders' perspective, like:

1.1 Client change CI

Design trends and information technologies are constantly evolving in the IT sphere. That's why often is necessary to change corporate identity or further developing something more. For example, that could be new logo, colour scheme or another UI component. Regular information like new company address, mail or telephone number also is possible alternative.

1.2 Show, hide or edit some existing components

When software is in working state, its owner wants to make visible or excludes some elements, even edits them and fill with new content.

1.3 Design trends changing constantly

Nowadays, to own successful mobile applications is required to follow up and to reflect in their user interface, the last UX/UI design tendencies and practices (Latest UX/UI tendencies, 2017; Unger and Boag, 2017).

1.4 We are people and make mistakes

We are human beings and sometimes we need to edit in the mobile application's content some details (text or graphic one).

For these different situations, the stakeholders like designers, developers, clients and customers need a mediator tool, which can control visual appearance of the application they use. This software should do that remotely. In that way, everybody has a possibility to access it from his/her personal computer, laptop or tablet.

After research based on (Stoeva, M., 2014; Atanasova and Malinova, 2016; Dzambazov, 2017) authors of this paper decide to develop such kind of tool. It has a possibility to control whole UI of the described down shopping mobile application and that action is accomplished remotely. Now authors are working on a project where this UI tool is used like a base of independent instrument, which has a possibility to control in easy way different kind of mobile apps, not only shopping one.

The current developed mediator tool follows and implements these important functionalities:

- Multiplatform functionality – described tool controls UI visual appearance of mobile software working on devices, working with different operating systems like iOS and Android.
- Multiple accounts functionality – this central backend tool provides different kind of user accounts access like super-administrators, administrators (for experience users) and guest (for nonexperience one).
- Safety design control – Users can add, edit or delete all visual elements in the mobile application UI, but per the current working design theme. Described tool helps them to avoid uploading of "wrong" graphic material via controlling mechanism, which check file type, size and not relevant visual elements. Thus, it protects mobile applications from nonconsistency in the corporate identity and crashes.
- Easy searching mechanism – The presented tool provides a possibility to search for specific UI components in the system. That ensures quick retrieval of needed information and speeds up design process.
- Measurement in the application functionality – Users have an overview on different metrics. In the tool reports customers, can see continuous tracking of the user like their interactions including success and error logs or other Google built-in Google Analytics, together with results from Facebook and Twitter social networks.

- Colour theme supporting functionality – The tool provides possibilities to change visual appearance of controlled mobile applications – these are all graphic and UI components.
- Notifications functionality – Administrators have a possibility to send on messages to all customers of controlled mobile software. That is so important contrivance when something will be or was changed in the remote application.

2 Motivation

The world becomes mobile, and so does the customers. Recently, it is no longer enough to make your own application which follow last UX rules (Unger and Chandler, 2009), but to you can control their UI remotely – without need to have a programing skills. The possibility for UI personalization is becoming the dominant feature of the mobile applications. The other pilot advantage is the level of software complexity. Large goods world help apps targeted at the reduction of complexity by selecting offers that meet the preferences of each consumer and fit into his current needs. Most of the UI tools are faced to prototyping and wireframing of software, and not to their generation. These are programs like Balsamiq, Uxpin UX Design Platform (Wireframing and prototyping tools, 2017), etc. But the current presented tool decides this problem easily. Except that, it controls UI design and helps users to plan and set push notifications, dynamically exchange videos and images, notification texts – even separately for each mobile device. This UI remote tool is also responsive (Unger and Chandler, 2009) and can be used on any device. It is flexible supporting software from backend type and is developed like a native terminal application for mobile devices, working on iOS and Android operating systems. The described solution can be visual fully customized.

For now, it is implemented like a mediator backend web solution, which can be connected to native mobile applications for iOS and Android. But its architecture allows to be additionally developed for UI of mobile applications built-in on other operating system platforms:

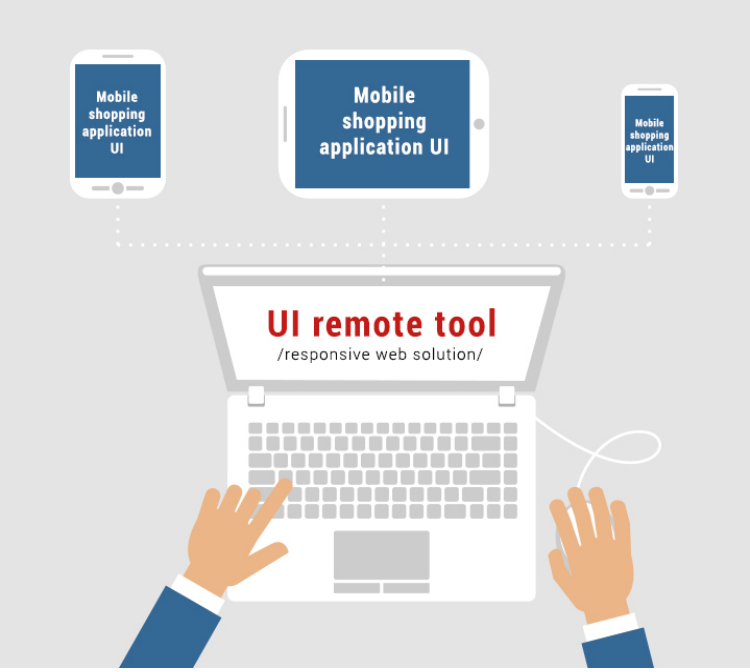

Figure 1. *UI Remote tool, which controls mobile native shopping application, installed on different mobile devices (smartphone and tablets). This solution is an instrument for mobile devices and can be used about any e-commerce web shop application. It consists of power backend and native terminal applications for smartphone and tablet.*

Development process of the UI remote tool was separated in two main parts. First one was its wireframing, which includes creation of all sketches for different applications screens and their behaviours. Second part relates to its architecture and real implementation. Both will be described in the next paper articles.

3 Software wireframing

In this section attention is turned on the UI tool software wireframing (Foster, 2017). This process is very important part from the whole implementation, because this is a stage of planning and clarifying all requirement of the mediator remote instruments. These are features like defining preliminary visual appearance and behaviour of the software. Making first mock-ups, but not real programing avoids mistakes from going to wrong directions. The UI remote tool includes the below described behaviour.

It saves customers' time for active maintenance of their content in the application; because it is connected via data feeds and refers to the product information and real-time availabilities directly from everywhere. For controlling these UI processes, the customer registers or logins to use its own account, but without the need leaving the shopping app.

This UI remote tool, which is from type backend, allows users to fully adapt mobile corporate design, so that the brand message is consistently transported across the whole application. It helps to plan mobile commerce campaigns. It prepares push notifications. It dynamically exchanges videos, banners, slideshows and individual controls in the installed mobile applications. It personalized notification texts – separately for each end-device. The whole tool goes through a couple of wireframing stages, because these possibilities are not easy to implement in a user-friendly way. In this paper, will be described only last one, which corresponds to the latest UX/UI practices (Atanasova and Malinova, 2016; Unger and Boag, 2017). All of them are created with the help of Balsamiq program (Wireframing and prototyping tools, 2017).

For the present, the UI remote tool is working with the mobile shopping application for iOS and Android operating systems. So, its specific functionality is defined from that business sphere and needs. Here are listed most important main wireframe screens, which control the visual appearance of mobile shopping application.

First important part of the UI remote tool is its main control panel.

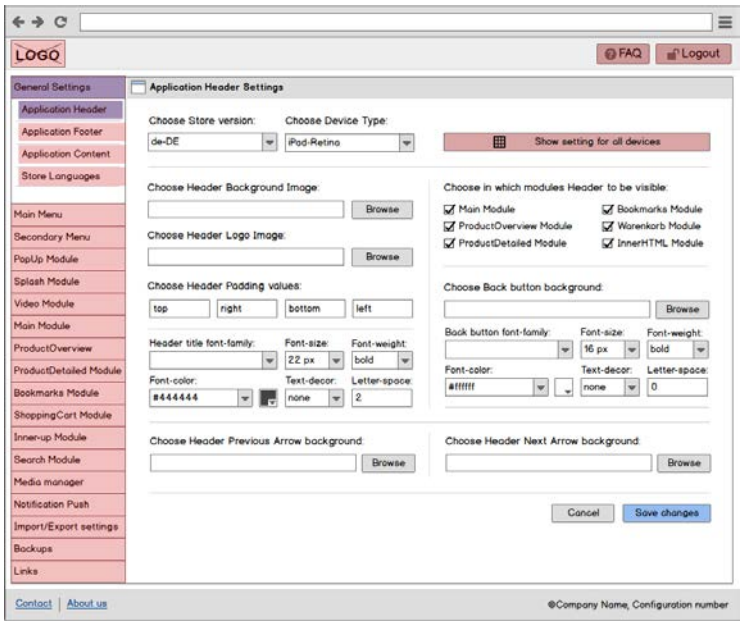

Figure 2. Main control panel

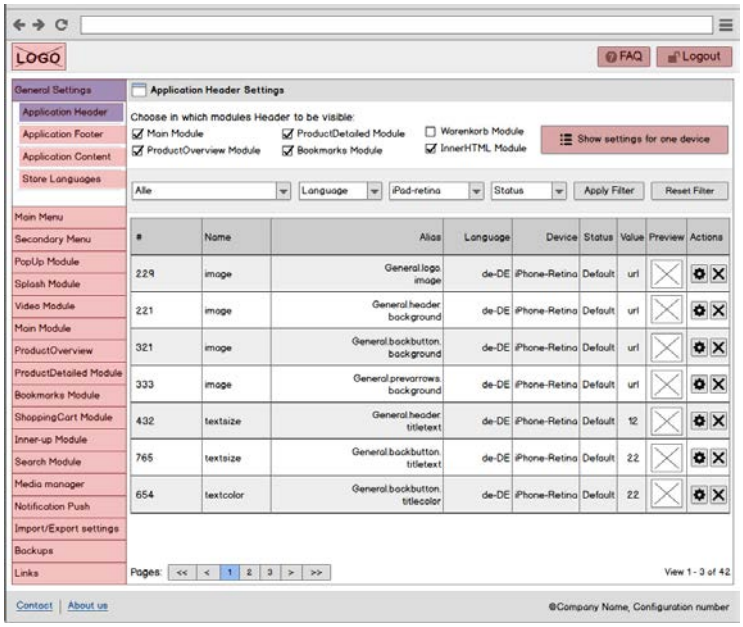

Figure 3. Main control panel settings in table view

On figure 2 are shown all different settings, which UI tool can control. Let's assume that the whole main interface is separated in four areas, which belong to the general settings section. These are application header, footer, content and languages. All of them can be repositioned in the controlled application screen. For example – header can be located on the bottom, but footer on the top. Different elements inside in the program structure are possible to be edited for every program language version. For example, customer can choose graphics on the header like background image or logo. Also, user can define distances between elements and their visibility on specific mobile modules or devices.

UI remote tool provides possibility to view these settings in a grid table view (figure 3).

Main menu and secondary menus are other vital modules, which can be controlled with UI tool in the mobile application. They are responsible for visualization of the main features belong to controlled mobile application. For example, to define whether will be or not available bookmark module via corresponding button in the menu.

Splash module defines preload module for preliminary loading. User has different possibilities like choose static image, video file in .mp4 file format or slideshow, consists from a couple of images (their number is defined again from the user).

Notification push is one of the interesting and useful functions. It takes separate corresponding section in the backend tool. This is the place where we can send some important messages to mobile users of shopping application; even they don't use application, at that moment. For that purpose, it is used standard service of Android and Apple devices, called Push Notification. *Push Notification* features was enabled in iOS 3.0. It is a way for one application to send information, even when the app isn't in use, to your phone (*via a badge, alert, or pop up message*). For example, mobile shopping app with enabled push notifications can send you the latest sale products in the moment, even if the app isn't running. The notification will send you messages, and when you touch it, it will direct return to the app for more information. UI tool users can decide for which operating system to be send, and when – now, scheduled or later with delay.

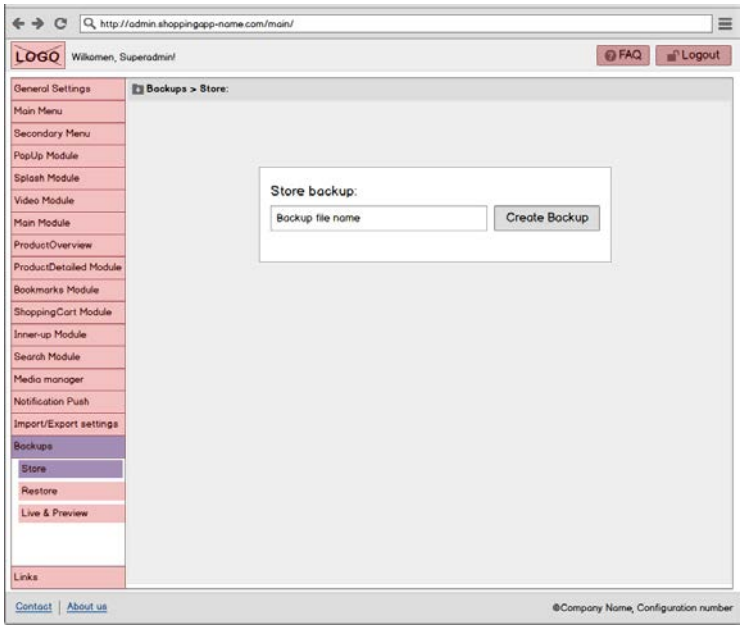

Figure 4. Backup screen

Import/Export section is responsible for setting new definitions, making changes on existing one or editing values, which cannot be controlled from the UI tool yet. This functionality is available with use of specific structured *.yml files (YAML, 2017), because the whole mobile user interface is programmed with YAML language. That is the big reason and advantage, which make UI tool so flexible. The software also allows users to generate and control back up (figure 4) of the whole mobile user interface states. That is very useful, if something wrong occurs, in the time of editing process or while preview changes before their live release.

All media elements, which are used in the user interface, are controlled via Media module. Inside that users can upload, edit, delete or download pictures and videos in different file formats like .jpg, .png, .gif and mp4. UI control tool gives possibility to manage external links and HTML pages, included in the mobile native application. They are controlled and are assigned via Inner-up module. All described functionalities of UI remote tool are available thankful to the flexible architecture and technologies, used for the implementation. Next section of this article is dedicated to them.

4 Software architecture and implementation

After the first part of the development was ready and all challenges were met, become a time to face up implementation. The whole system of this web solution is divided into two parts:

- Preview system (preview) – Designed for developers. This system pre-configures the mobile application and is used for testing purposes.
- Live system (live) – Designed for end users. Settings for this system are imported from Preview system.

The presented tool is firstly created with PHP Symfony2 version 2.1.7, together with Bootstrap 3.3 frameworks (PHP Symfony, Bootstrap 2017). Now is upgraded to PHP Symfony3. These chosen technologies allowed authors to implement this solution with three-layered architecture and easily adapted to control user interface of different kind mobile applications, described here in details.

The architecture itself is defined from PHP Symfony. Therefor it uses Model View Controller (MVC) pattern (MVC and MVP Patterns, 2017). This pattern is a user interface presentation that focuses on separating the user interface (View) from its business layer (Model). In the case of the UI remote tool

visual appearance is described in .YML files, but functionality are implemented in php files. The pattern separates responsibilities across three components:

- View, which is responsible for rending all interface elements, including all html files, without images and other media and .css files;
- Model, which is responsible for business behaviors and state management. It consists from .yml and datafeed files;
- Controller – for responding to UI actions and all tools' logic. It processes the user requests, get appropriate data and output the requested design view.

All three components can directly interact with each other in the described implementation.

UI tool responsiveness is ensured from Bootstrap – one of the most famous HTML, CSS, and JavaScript framework (PHP Symfony, Bootstrap 2017) for developing adaptive and mobile first projects. Module based functionality is ensured through components origin of PHP Symfony. But the whole UI tool project specifics consist from two main groups of bundles. First one is not a part from Symphony framework, but further developed from the team. It consists following modules:

- Backups module, used for archive of back-up files;
- Certificates module, contains files with certificates for mobile application;
- ImportDataFeed module, contains data feed scripts for importing;
- Datafeed module, is necessary for storage of actual import files;
- Module export, is responsible for storage of temporary export files and zip archives;
- Module files, is designed to storage files, which provide access to mobile devices;
- Module import, is designed to storage of files to import settings.

Second group contains modules, which represents installed third-party components. These are:

- KnpPaginatorBundle, which is responsible for paging navigation in the tables with settings;
- RMSPushNotificationsBundle, this component is sending Push Notifications for mobile devices.

UI remote tool visual appearance can be seen in the next figure (figure 5). Left side shows the functional screen (table view) for controlling splash screen elements of the user interface. These are editable components like logo graphic, application language, colour code of the preload component (text and strip), device type, visibility, preload video, general alignment and position. In one sentence: every row of the table presents one property, described with unique id, name, identifier, language version device type. These values can be edited or turned on/off to default state via clicking Reset button in the Actions' column. For example, if user wants to modify a property value, first he must click Edit button. After that, it will appear window pop-up, like the one, shown on the right side of figure 5. After all changes are made, user must click "Save" changes button for saving new values.

UI remote tool suggests powerful mechanism for additional protection of controlled mobile application. It is accomplished by the following:

When administrator is ready with all desired changes and wants to reflect them on the live software, first he/she should go to the Backups section. There will find three sub sections: Store, Restore and Live and Preview versions. "Store" part is very important section, because in that place users make backup copy of the current mobile application state. This action must be executed before every change in the tool and reflects in the app!!! In that way is ensured, that if the system crashes for some reason, will be available a fresh copy! Live and Preview section is the place where user can get all changed values from UI tool and applies to the Preview application version. By clicking Set to Live button will be assigned all preview settings to the "Live" version of the mobile application.

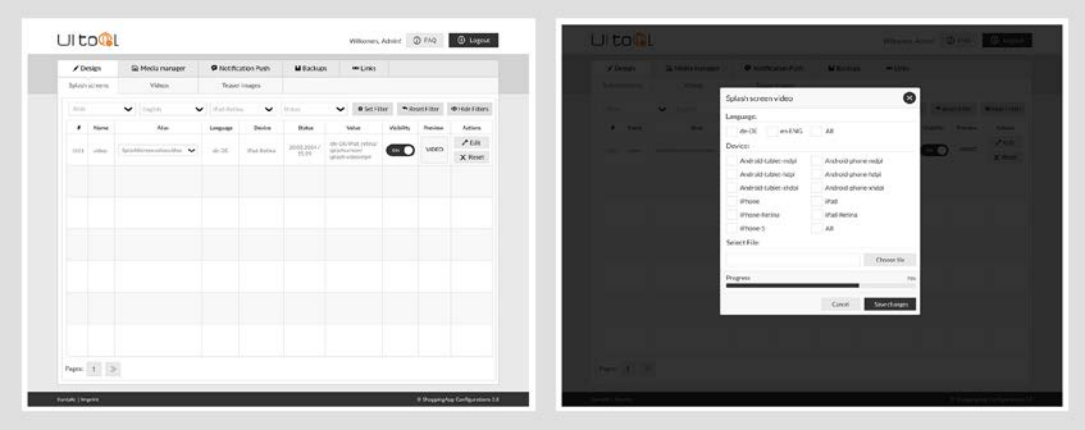

Figure 5. User interface of the UI remote tool

Let's assume in the splash screen section of the UI tool is set new colour value for preload strip, text and logo (see figure 5). After that, it is clicked *Set to Live* button in the corresponding section. All changes will be reflected in the mobile application, how is shown in figure 6 (changing orange to blue *from left to right part*).

Figure 6. Results on the mobile application's screens for tablet and smartphone

5 Conclusion

Current paper presents a flexible web solution. It can dynamically control visual interface of the multiplatform mobile application. The status of described web tool is in the working mode. It was completely released and is using from two end-users. Now it is developing an update for Windows 8 mobile and edition of Windows 10 operating system. Also, authors are working on new feature for converting video with web services.

After their finalization, next purpose will be to adapt this mediator solution to be more independent from the controlled mobile application type. UI tool will have a possibility to manage easier visual appearance of different mobile apps.

References

Stoeva, M. (2014), "Interactive Multimedia Tool for Dynamic Generation of Web Interfaces with HTML5/PHP/MySQL and JavaScript", *International Journal of Emerging Technology & Advanced Engineering*, Vol. 4, Iss. 9, p. 412–418, ISSN: 2250-2459, ISO 9001:2008 Certified Journal.

Unger, R., Chandler, C. (2009), "A Project Guide to UX Design: For user experience designers in the field or in the making", New Riders, ISBN-13 978-0-321-60737-9.

Foster, N. (2017), *UX/UI prototyping/wireframing tools*, URL: http://www.creativebloq.com/wireframes/top-wireframing-tools-11121302 (visited on 17/06/2017).

Atanasova, M., Malinova, A. (2016), "User interfaces in business information systems – problems and solutions", Plovdiv University "Paisii Hilendarski", Faculty of Mathematics and Informatics, Scientific Conference Innovative ICT in Business and Education: Future Trends, Applications and Implementation, Pamporovo, ISBN: 978-954-8852-72-2, p. 29-40.

Latest UX/UI tendencies, 2017, URL: https://www.invisionapp.com/blog/ux-design-trends-2017, https://blogs.adobe.com/creativecloud/the-future-is-now-10-design-predictions-for-2017/, https://www.quora.com/What-will-be-the-biggest-UX-UI-design-trends-in-2017 (visited on 17/06/2017).

Unger, R., Boag, P. (2017), "Experience Revolution", Smashing Media AG, Freiburg, Germany, ISBN: 978-3-945749-54-8 (Kindle).

Stoeva, M. (2014), "Specifics in design and adaptation of interactive user interfaces for mobile applications", *Jubilee conference for 125 anniversary of mathematics and natural Sciences*, Sofia University "St. Kliment Ohridski" Sofia, Bulgaria, 76-78 p., ISSN 1313-9045.

Dzambazov, V. (2017), *Tendencies in user interfaces evolution*, URL: http://www.journal-informo.eu/content/012/003.pdf, 2014 (visited on 17/06/2017).

Wireframing and prototyping tools, 2017, *Expression Blend*, URL: http://www.microsoft.com, *Balsamiq* URL: http://balsamiq.com, *Mockflow* URL: www.mockflow.com, *Moqups* URL: https://moqups.com, UXPIN URL: https://www.uxpin.com, (visited on 17/06/2017).

YAML (Yet Another Markup Language) 2017, URL: http://www.yaml.org/start.html, (visited on 17/06/2017).

MVC and MVP Patterns, 2017, URL: https://www.infragistics.com/community/blogs/todd_snyder/archive/2007/10/17/mvc-or-mvp-pattern-whats-the-difference.aspx, (visited on 17/06/2017).

PHP Symfony development framework, 2017, URL: http://symfony.com, (visited on 17/06/2017).

Bootstrap framework, 2017, URL: http://getbootstrap.com, (visited on 17/06/2017).